1. **What is RDBMS?**
   The software used to store, manage, query, and retrieve data stored in a relational database is called a relational database management system (RDBMS).

2. **What is SQL?**
   SQL is Structured Query Language, which is a computer language for storing, manipulating and retrieving data stored in relational database.

3. **Write SQL Commands.**

   DDL – Data Definition Language
   DML – Data Manipulation Language
   DCL – Data Control Language
   DQL – Data Query Language

4. **What is join?**
   JOIN is used to combine the results of two tables. To perform a join, each of the tables must have at least one field which will be used to find matching records from the other table. The join type defines which records will go into the result set.
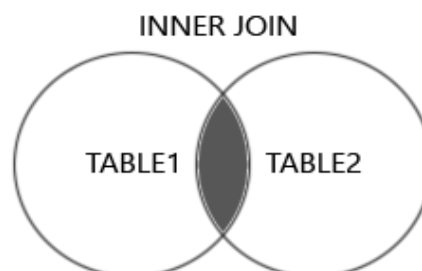
5. **Write type of joins.**
   - **Inner join:** returns rows when there is a match in both tables.

     EX: select * from [table1] inner join [table2]
                         On
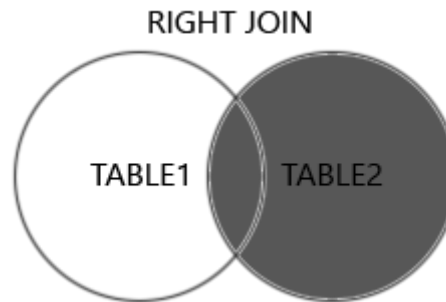   table1.matching_column = table2.matching_column;

INNER JOIN
TABLE1   TABLE2

- **Right join:** Returns all rows from the right table, even if there are no matches in the left table**.**
  EX:
  select * from [table1] Right join [table2]
                  On
  table1.matching_column = table2.matching_column;
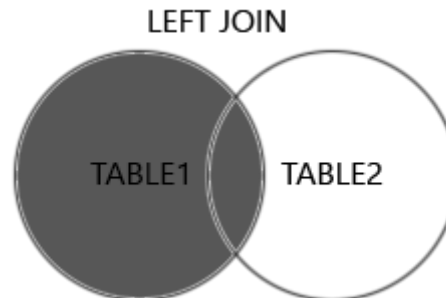
RIGHT JOIN

TABLE1   TABLE2

- **Left join:** Returns all rows from the left table, even if there are no matches in the right table.
  EX:
  select * from [table1] Left join [table2]
                  On
  table1.matching_column = table2.matching_column;

LEFT JOIN

TABLE1   TABLE2
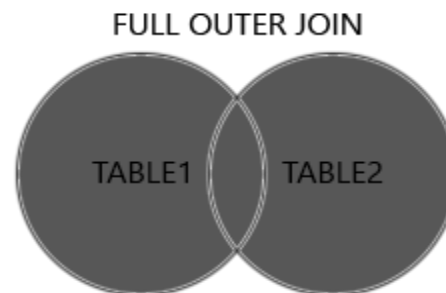
- **Full join:** returns rows when there is a match in one of the tables.
  EX:
  select * from [table1] full join [table2]
                  On
  table1.matching_column = table2.matching_column;

FULL OUTER JOIN

TABLE1   TABLE2

6. **What is API Testing?**
API (Application Programming Interface) is a computing interface which enables communication and data exchange between two separate software systems.
- Another definition, API (Application Programming Interface) is a computing interface which enables communication and data exchange between two separate software systems.
- The purpose of API Testing is to check the functionality, reliability, performance, and security of the programming interfaces.

7. **Types of API Testing.**
There are 3 types of API testing.
   1) **Open APIs:** These types of APIs are publicly available to use like OAuth APIs from Google. It has also not given any restriction to use them. So, they are also known as Public APIs.
   2) **Partner APIs:** Specific rights or licenses to access this type of API because they are not available to the public.
   3) **Internal APIs:** Internal or private. These APIs are developed by companies to use in their internal systems. It helps you to enhance the productivity of your teams.

8. **What is Responsive Testing?**
A responsive web design involves creating a flexible web page that is accessible from any device, starting from a mobile phone to a tablet. A responsive web design improves users' browsing experience.
   - ➤ Some points to be understand for responsive testing
      - How website testing differs from a mobile device to a computer
      - The challenges involved in testing a responsive website
      - Rules and guidelines to be followed during responsive design testing.
      - Lastly, various tools available to perform responsive testing.

9. **Which types of tools are available for Responsive Testing?**
   - LT Browser
   - Lembda Testing
   - Google Resizer
   - I am Responsive
   - Pixel tuner

10. **What is the full form of .ipa, .apk?**
    .ipa: iOS package App Store
    .apk: Android Package

11. **Difference between RDBMS vs DBMS.**
    RDBMS:
    - RDBMS stores data in tabular form.
    - Multiple data elements can be accessed at the same time.
    - Data is stored in the form of tables which are related to each other
    - Normalization is present.
    - RDBMS supports distributed database.
    - It uses a tabular structure where the headers are the column names, and the rows contain corresponding values.
    - It deals with large amount of data.
    - Keys and indexes do not allow Data redundancy
    - It is used to handle large amount of data.
    - All 12 Codd rules are satisfied.
    - More security measures provided.
    - It supports multiple users.
    - Data fetching is fast because of relational approach.
    - There exist multiple levels of data security in a RDBMS.
    - Higher software and hardware necessities.
    - Examples: MySQL, PostgreSQL, SQL Server, Oracle, Microsoft Access etc.

    DBMS:
    - DBMS stores data as file**.**
    - Data elements need to access individually.
    - No relationship between data.
    - Normalization is not present.
    - DBMS does not support distributed database.
    - It stores data in either a navigational or hierarchical form.
    - It deals with small quantity of data.
    - Data redundancy is common in this model.
    - It is used for small organization and deal with small data.
    - Not all Codd rules are satisfied.

- Security is less.
- It supports single user.
- Data fetching is slower for the large amount of data.
- The data in a DBMS is subject to low security levels with regards to data manipulation.
- Low software and hardware necessities.
- Examples: XML, Window Registry, Forxpro, dbaseIIIplus etc.

12. How Many constraint and describes itself.

Primary Key: A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table.

EX:

```
CREATE TABLE Student
(
  ID int(6) NOT NULL UNIQUE,
  NAME varchar(10),
  ADDRESS varchar (20),
  PRIMARY KEY(ID)
);
```

Default: Sets a default value for a column if no value is specified.

EX:

```
CREATE TABLE Student
(
   ID int(6) NOT NULL,
   NAME varchar(10) NOT NULL,
   AGE int DEFAULT 18
  );
```

Unique: Ensures that all values in a column are different.

EX:

```
CREATE TABLE Student
(
  ID int(6) NOT NULL UNIQUE,
  NAME varchar(10),
  ADDRESS varchar(20)
 );
```

Check: Ensures that the values in a column satisfies a specific condition.
EX:

CREATE TABLE Student
(
  ID int(6) NOT NULL,
  NAME varchar(10) NOT NULL,
  AGE int NOT NULL CHECK (AGE >= 18)
);

Not Null: Ensures that a column cannot have a NULL value
EX:

CREATE TABLE Student
(
  ID int(6) NOT NULL,
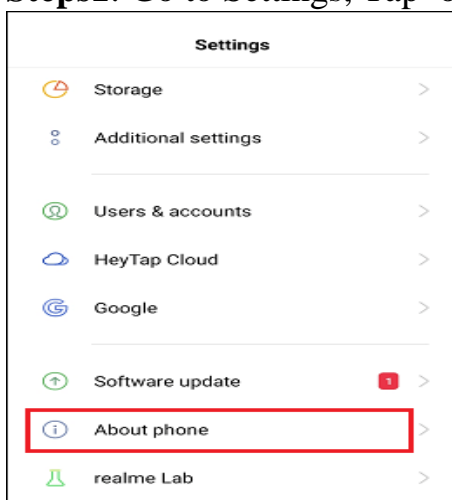 NAME varchar(10) NOT NULL,
  ADDRESS varchar(20)
);

**Foreign key:** Prevents actions that would destroy links between tables.
EX:

CREATE TABLE Orders
(
  O_ID int NOT NULL,
  ORDER_NO int NOT NULL,
  C_ID int,
  PRIMARY KEY (O_ID),
  FOREIGN KEY (C_ID) REFERENCES Customers(C_ID)
)

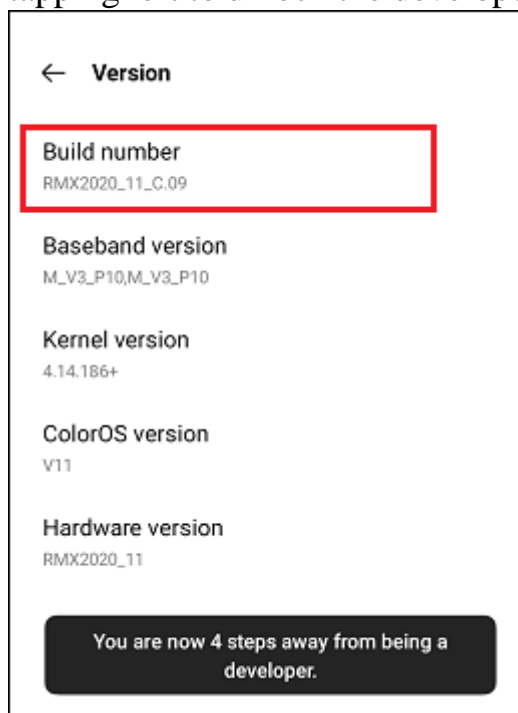13. **How to create step for to open the developer option mode ON?**

Steps1: Go to Settings, Tap on the " About phone" option.

**Steps2:** Tap on the Android version option to find "Build Number".
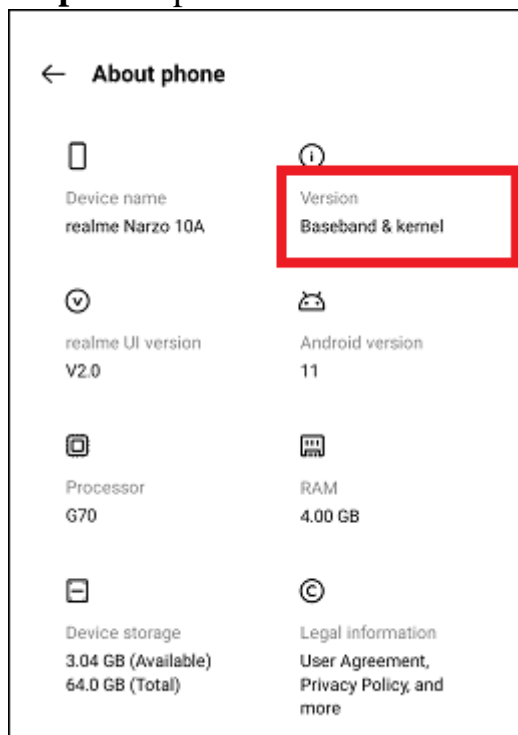
Steps3: Tap on "Build Number" seven times continuously to enable developer options. While tapping on it, you will see a counting number for tapping left to unlock the developer options.

Steps4: Once the Developer options get enabled, a toast message appears, showing that you are now a developer.

Steps5: Go back to the "Settings" screen, and under "Additional Settings," you will find Developer options.
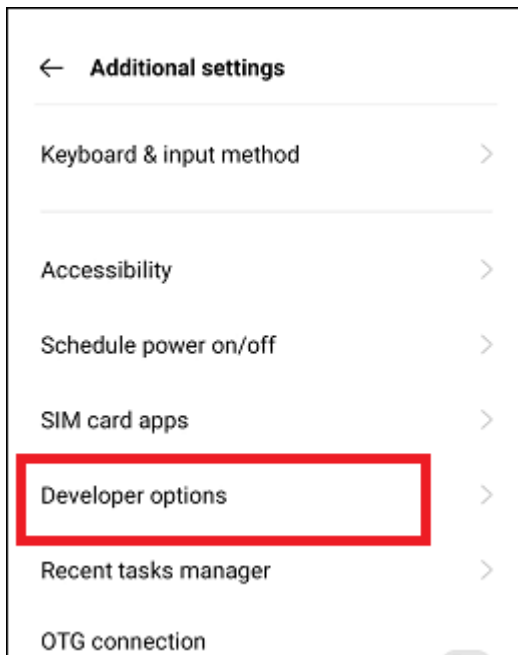
Steps6: Tap on Developer options and enable the toggle button next to Developer options.